Queue

Objective

* To gain understanding of how an implementation of an ADT is used by an application program.
* To learn how Queues can be implemented to perform its operations
* Implementation of Queue using array and linked list both.

Task

1. Array based circular Queue
2. Linked List based Queue
3. Joseph problem (post Lab work)

Problems

1. Implement Queue using array. Complete the following methods in Queue class.

interface Queue<E>{

public boolean isEmpty();

public boolean isFull();

public void enQueue(E value)throws FullQueueException;

public E deQueue()throws EmptyQueueException;

}

class ArrayQueue<E> implements Queue<E>{

public static final int MAX=4;

int Rear=MAX-1;

int Front=MAX-1;

E QueueList[];

ArrayQueue(){

this(MAX);

}

ArrayQueue(int size){

QueueList= (E[])new Object[size];

}

public boolean isEmpty(){

}

public boolean isFull(){

}

public void enQueue(E value)throws FullQueueException{

}

public E deQueue()throws EmptyQueueException{

}

}//End of Class ArrayQueue

1. Implement stack using linked list. Complete the methods of LinkedQueue class.

class QueueNode

{

object info;

StackNode next;

public QueueNode(){}

public QueueNode(Object j, QueueNode p)

{info = j; next = p;

}

}

public class LinkedQueue

{

QueueNode Rear, Front;

public LinkedStack() {

Rear = null;

Front=null;

}

public boolean isEmpty(){

}

public boolean isFull() {

}

public int deQueue() {

}

public void enQueue(int data) {

}

}

1. Joseph Problem

In the children's game "hot potato," a group of *n* children sit in a circle passing an object, called the "potato," around the circle. The potato begins with a starting child in the circle, and the children continue passing the potato until a leader rings a bell, at which point the child holding the potato must leave the game after handing the potato to the next child in the circle. After the selected child leaves, the other children close up the circle. This process is then continued until there is only one child remaining, who is declared the winner. If the leader always uses the strategy of ringing the bell after the potato has been passed *k* times, for some fixed value *k*, then determining the winner for a given list of children is known as the ***Josephus problem***.

Solving the Josephus Problem Using a Queue

We can solve the Josephus problem for a collection of *n* elements using a queue, by associating the potato with the element at the front of the queue and storing elements in the queue according to their order around the circle. Thus, passing the potato is equivalent to dequeuing an element and immediately enqueuing it again. After this process has been performed *k* times, we remove the front element by dequeuing it from the queue and discarding it.